

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту
Завідувач кафедри _____ О.О. Ємець
(підпис)
« _____ » _____ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ТРЕНАЖЕРА
«ВИКОРИСТАННЯ ФУНКЦІЇ FIRST ПРИ ПОБУДОВІ
ПРЕДИКТИВНОГО АНАЛІЗАТОРА» ДИСТАНЦІЙНОГО
НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Франко Вадим Анатолійович

_____ « _____ » _____ 2021р.
(підпис)

Науковий керівник доц., каф.-мат. наук, Черненко О.О.

_____ « _____ » _____ 2021р.
(підпис)

ПОЛТАВА 2021р

РЕФЕРАТ

Записка: 44 стор., в т.ч. основна частина 41 стор., джерел – 10.

Мета бакалаврської роботи – алгоритмізація та програмування елементів тренажера «Використання функції FIRST при побудові предиктивного аналізатора» дистанційного навчального курсу «Теорія програмування».

Предмет розробки — алгоритм та програмна реалізація роботи тренажеру з теми «Використання функції FIRST при побудові предиктивного аналізатора».

Методи, які були використані для розв’язування задачі –

Середовище розробки IDE Visual Studio;

Платформа Unity.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 4 |
| РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ..... | 6 |
| РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД..... | 8 |
| РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА | 10 |
| 3.1. Теоретичний матеріал з теми..... | 10 |
| 3.2. Алгоритмізація задачі за темою роботи | 14 |
| 3.3 Блок-схема алгоритму тренажера..... | 24 |
| РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА | 25 |
| 4.1 Опис програмної реалізації..... | 25 |
| 4.2 Інструкція для роботи з тренажером..... | 30 |
| ВИСНОВКИ..... | 40 |
| СПИСОК ЛІТЕРАТУРИ..... | 42 |
| ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ | 44 |

ВСТУП

Дистанційний метод навчання є більш зручним і значно дешевшим, ніж традиційний, завдяки ефективному використанню навчальних приміщень, полегшеному коригуванню електронних навчальних матеріалів та мультидоступу до них.

Широкий доступ студентів до освітніх ресурсів, не обмежені часом та відстанню можливості працювати з електронними варіантами програмно-методичного забезпечення по тим чи іншим дисциплінам формують відповідний рівень предметного діалогу «викладач-студент».

Запровадження в навчальний процес нових технологій навчання значно підвищує відповідальність викладача за якість навчально-методичних матеріалів, зміст яких потребує постійного поновлення і вдосконалення, а досвід традиційної організації навчального процесу – адаптації до нової системи комунікацій між викладачем і студентом. Застосування автоматизованих систем контролю знань, консультування і навчання за допомогою комп'ютерних технологій дозволяє істотно скоротити витрати часу викладача на етапах контролю знань, зняти перенапругу та нервозність студентів в процесі навчання та складання іспитів, зробити процес навчання більш яскравим і привабливим. За таких умов викладач виступає не тільки розробником електронного варіанту навчально-методичного забезпечення своєї дисципліни, а і керівником пізнавальних процесів. Оперативний зворотній зв'язок із студентами здійснюється за допомогою результатів тестової перевірки знань.

Мета бакалаврської роботи – алгоритмізація та програмування елементів тренажера «Використання функції FIRST при побудові предиктивного аналізатора» дистанційного навчального курсу «Теорія програмування».

Предмет розробки — алгоритм роботи тренажеру з теми «Використання функції FIRST при побудові предиктивного аналізатора».

Завдання бакалаврської роботи - алгоритмізація та програмна реалізація тренажера.

Структура пояснювальної записки до бакалаврської роботи: у першому розділі розглянуто постановку задачі; у другому розділі представлено інформаційний огляд; у третьому розділі представлено теоретичний матеріал з теми, алгоритмізацію задачі за темою роботи; у четвертому розділі представлено процес програмної реалізації тренажеру.

У додатку міститься диск з матеріалами проекту.

Обсяг пояснювальної записки: 44 стор., в т.ч. основна частина - 41 стор., додатки - 1 стор., джерела - 10 назв.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Основним завданням роботи є алгоритмізація елементів тренажера «Використання функції FIRST при побудові предиктивного аналізатора» дистанційного навчального курсу «Теорія програмування».

Розглянемо основні завдання роботи:

- розглянути теоретичні відомості про використання функції FIRST при побудові предиктивного аналізатора;
- розглянути перспективи дистанційного навчання;
- провести вибір та обґрунтування мови програмування;
- розробити алгоритм тренажера;
- розробити навчальний тренажер для закріплення знань та навичок.

Тренажер повинен передбачити можливість перевірки правильності кожного з кроків алгоритму та повідомляти про помилку у разі неправильної відповіді.

При проходженні тренажеру спочатку користувачеві повинні відображатися теоретичні завдання, а вже потім приклади.

При розробці програмного продукту пропонується використати наступну структуру:

- стартова сторінка:
 - тема тренажера;
 - інформація про розробника;
 - інформація про керівника;
 - можливість переходу до тестування;
- теоретичні завдання:
 - питання;
 - варіанти відповіді;
 - кнопка для переходу до наступного питання;
- приклади:
 - умова;

- спосіб розв'язання;
 - кнопка для переходу до наступної умови;
- результат тестування:
 - повідомлення про завершення;
 - можливість повторного проходження
 - кнопка для виходу.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

Система освіти пов'язана з реалізацією процесу поширення знань та є найважливішим компонентом життя, оскільки охоплює діяльність сформованих суспільних інститутів, які здійснюють підготовку молоді до життя на основі отриманих знань в дошкільних дитячих установах, в середніх спеціальних, професійно-технічних і вищих навчальних закладах.

Значення освіти в житті людини постійно зростає, а нагальна потреба в постійному підвищенні рівня знань призводить до появи в суспільному житті поняття безперервності освіти. Цей процес супроводжується пошуком нових форм і методів навчання, що сприяють правильному формуванню наукового світогляду та глибокому розумінню економічних та соціальних змін в житті суспільства.

У світі продовжує відбуватися інформаційна революція, що актуалізує проблеми модернізації освіти. Тому державна політика у сфері вищої освіти має ґрунтуватися на принципах: сприяння сталому розвитку суспільства шляхом підготовки конкурентоспроможного людського капіталу, створення умов для освіти протягом життя та доступності вищої освіти. Формування і реалізація державної політики у сфері вищої освіти забезпечуються шляхом гармонійної взаємодії національних систем освіти, науки, бізнесу та держави з метою забезпечення стійкого соціально-економічного розвитку держави та розширення можливостей для здобуття вищої освіти та освіти протягом життя.

Політика модернізації освітніх систем розвинених країн усе більше орієнтується на розвиток дистанційної освіти. Сучасні тенденції в освіті вимагають підготовки фахівців, які здатні до професійної та інноваційної діяльності, оновлення знань, проектування особистісного та професійного зростання. Дистанційне навчання – це прогресивна педагогічна технологія, яка ґрунтується на сучасних досягненнях у галузі інформаційних та телекомунікаційних технологій. Впровадження дистанційного навчання

допоможе реалізувати принцип навчання впродовж життя, а отже забезпечить безперервність освіти.

Популярність дистанційної освіти на основі інтернет-технологій, особливо в системі вищої освіти, щорічно зростає, що пов'язано з перевагами, які надає така форма навчання, такими, як можливість отримувати освіту без відриву від виробництва, економія ресурсів та часу, розширенням сфери додаткової освіти тощо.

З огляду на світові тенденції зростання ролі дистанційної освіти, можна стверджувати, що цей розвиток матиме дедалі зростаючий вплив на забезпечення національних інтересів [2].

Дистанційна форма навчання в Україні почала впроваджується понад десять років тому. Сучасні завдання, які поставлені перед системою освіти України вимагають створення такої системи освіти, яка забезпечуватиме перехід до безперервної освіти. Зволікання з розвитком дистанційної освіти загрожує зниженням конкурентоспроможності української освіти у світовому просторі.

Система дистанційного навчання в Україні наразі перебуває лише на стадії становлення, але за умови використання світового досвіду, поєднання прогресивних технологій дистанційної освіти з кращими технологіями та методами класичних форм навчання, в перспективі вона може набути стрімкого розвитку. Але широке впровадження і розвиток дистанційної освіти в Україні потребує вирішення комплексу завдань за такими напрямками, як управлінсько-організаційне забезпечення; матеріально-технічне та фінансове забезпечення; кадрове забезпечення потреб дистанційної освіти; методичне забезпечення з урахуванням специфіки дистанційного навчання; просування дистанційної освіти на освітньому ринку та ринку праці.

Отже, на сьогодні дистанційна освіта в Україні перебуває на етапі впровадження і використання в освітньому процесі вищих навчальних закладів. Система дистанційного навчання не замінить традиційну систему вищої освіти, а доповнюватиме її, і, водночас, матиме вплив на розвиток освіти як в Україні так і в усьому світі, та удосконалення української системи вищої освіти [3].

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Теоретичний матеріал з теми

«Предиктивний синтаксичний аналізатор – це синтаксичний аналізатор, що працює методом рекурсивного спуску без повернень (відкотів). Це можливо, якщо з граматики видалена ліва рекурсія і вона лівофакторизована.»

«У багатьох випадках акуратне розроблення граматики, видалення з неї лівої рекурсії та її ліва факторизація дозволяють одержати граматику, яка може бути проаналізована синтаксичним аналізатором, що використовує метод рекурсивного спуску і не потребує відкоту (предиктивним аналізатором).»

«Нерекурсивний предиктивний аналізатор можна побудувати за допомогою явного використання стека замість неявного при рекурсивних викликах. Ключова проблема предиктивного аналізу полягає у визначенні продукції, яку потрібно застосувати до нетермінала. Для пошуку продукції може бути використана таблиця розбору.»

Модель предиктивного синтаксичного аналізатора, що керується таблицею, показана на рис. 3.1.



Рисунок 3.1 - Схема предиктивного аналізатора

«Аналізатор має вхідний буфер, стек, таблицю розбору і вихідний потік. Буфер містить вхідний рядок, за яким йде правий кінцевий маркер \$ – ознака кінця рядка. Стек містить послідовність символів граматики з \$ на дні. Спочатку стек містить початковий символ граматики безпосередньо над

символом $\$$. Таблиця розбору – це двовимірний масив $M(A,a)$, де A – нетермінал; a – термінал або символ $\$$.»

«Аналізатор керується програмою, що працює у такий спосіб. Програма розглядає X – символ на вершині стека і a – поточний вхідний символ. Ці два символи визначають дію аналізатора. Є три можливості.»

1. Якщо $X=a=\$$, аналізатор зупиняється і повідомляє про успішне закінчення розбору.
2. Якщо $X=a\neq\$,$ аналізатор видаляє X зі стека і просуває покажчик вхідного потоку на наступний символ.
3. Якщо X – нетермінал, програма розглядає запис $M(X,a)$ з таблиці розбору M . Цей запис являє собою або X -продукцію граматики, або запис про помилку. Якщо, наприклад, $M(X,a)=\{X\rightarrow UVW\}$, аналізатор замінює X на вершині стека на WVU (з U на вершині). Будемо вважати, що аналізатор на виході просто друкує використану продукцію. Якщо $M(X,a)=error$, аналізатор викликає підпрограму аналізу помилок.

«Алгоритм 1. Нерекурсивний предикативний аналіз»

Спочатку аналізатор знаходиться у конфігурації з $\$S$ (на вершині – стартовий символ S граматики G) і рядком $w\$$ у вхідному буфері.

Установити покажчик iP на перший символ $w\$$;

repeat

Позначимо через X символ на вершині стека, а через a – символ, на який вказує iP .

if X – термінал або $\$$ **then**

if $X=a$ **then** видалити X зі стека і перемістити iP

else *error*

else if $M(X,a)=X\rightarrow Y_1Y_2\dots Y_k$ **then** /* X – нетермінал*/

begin

видалити X зі стека;

```

    помістити у стек  $Y_k, Y_{k-1}, \dots, Y_1$  на вершині стека;
    вивести продукцію  $X \rightarrow Y_1 Y_2 \dots Y_k$ 
end
else error /*вхід таблиці  $M$  порожній*/
until  $X = \$$ . /*стек порожній*/

```

При побудові предиктивного аналізатора корисними виявляються дві функції, зв'язані з граматиною G . Ці функції, FIRST і FOLLOW, дозволяють побудувати таблицю предиктивного розбору для G , якщо, звичайно, це можливо. Множини, що породжуються цими функціями, можуть, крім того, бути використані при відновленні після помилок.

Нехай a – довільний рядок символів граматики. Функція $\text{FIRST}(a)$ – це множина терміналів, з яких починаються рядки, виведені з a . Якщо $a \Rightarrow^* \varepsilon$, то ε також належить $\text{FIRST}(a)$. Для побудови $\text{FIRST}(X)$ для всіх символів X граматики застосуємо такий алгоритм.

Алгоритм 2. Побудова множини FIRST для символів граматики.

Крок 1. Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$.

Крок 2. Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.

Крок 3. Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 Y_2 \dots Y_{i-1} \Rightarrow \varepsilon$. Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i = 1, 2, \dots, k$, то додати ε до $\text{FIRST}(X)$.

Наприклад, усе, що належить $\text{FIRST}(Y_1)$ належить також і $\text{FIRST}(X)$. Якщо з Y_1 не виводиться ε , то ми нічого більше не додаємо до $\text{FIRST}(X)$, але якщо $Y_1 \Rightarrow \varepsilon$, то додаємо $\text{FIRST}(Y_2)$ і т.д.

Тепер FIRST для будь-якого рядка $X_1 X_2 \dots X_n$ можна обчислити у такий спосіб. Вважаємо $\text{FIRST}(X_1 X_2 \dots X_n) = \{\}$. Додамо до $\text{FIRST}(X_1 X_2 \dots X_n)$ усі не ε -символи $\text{FIRST}(X_1)$. Додамо також всі не ε -символи з $\text{FIRST}(X_2)$, якщо

$\varepsilon \in \text{FIRST}(X_1)$, всі не ε -символи з $\text{FIRST}(X_3)$, якщо ε належить як $\text{FIRST}(X_1)$, так і $\text{FIRST}(X_2)$, і т.д. Нарешті, додамо ε до $\text{FIRST}(X_1 X_2 \dots X_n)$, якщо для всіх i $\text{FIRST}(X_i)$ містить ε .

Функція $\text{FOLLOW}(A)$ для нетермінала A - це множина терміналів a , що можуть з'явитися безпосередньо праворуч від A у деякій сентенціальній формі, тобто множина терміналів a таких, що існує породження вигляду $S\alpha \Rightarrow Aa\beta$ для деяких α і β . Відзначимо, що між A і a у процесі виведення можуть з'явитися нетермінальні символи, з яких виводиться ε . Якщо A може виявитися крайнім правим символом деякої сентенціальної форми, то $\$$ належить $\text{FOLLOW}(A)$.»

Приклад. Розглянемо граматику арифметичних виразів:

$$E \rightarrow TE', E' \rightarrow +TE' \mid \varepsilon,$$

$$T \rightarrow FT', T' \rightarrow *FT' \mid \varepsilon,$$

$$F \rightarrow (E) \mid id$$

Запишемо функції FIRST .

Для неї:

$$\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{ (, id \};$$

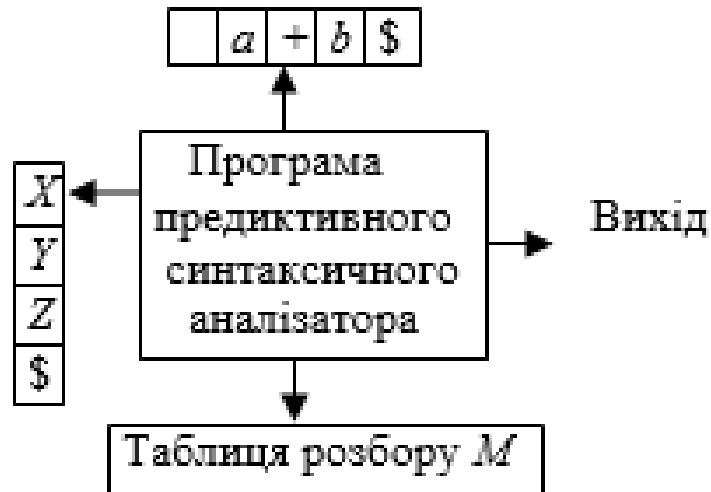
$$\text{FIRST}(E') = \{ +, \varepsilon \};$$

$$\text{FIRST}(T') = \{ *, \varepsilon \};$$

Наприклад, **id** і ліва дужка додаються до $\text{FIRST}(F)$ на кроці 3 алгоритму для FIRST , оскільки $\text{FIRST}(\text{id}) = \{\text{id}\}$ і $\text{FIRST}('(') = \{ (\}$ у відповідності з кроком 1. На кроці 3 при $i=1$ відповідно до продукції $T \rightarrow FT'$ до $\text{FIRST}(T)$ додаються також **id** і ліва дужка. На кроці 2 у $\text{FIRST}(E')$ включається ε [4, 5].

3.2. Алгоритмізація задачі за темою роботи

Крок 1. Виводиться пояснення: «Модель предиктивного синтаксичного аналізатора, що керується таблицею, показана на рисунку



».

Перехід на наступний крок.

Крок 2. Виводиться завдання: «Перелічіть компоненти аналізатора».

Варіанти відповіді:

- вхідний буфер;
- таблиця розбору;
- стек;
- вихідний потік.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Аналізатор має вхідний буфер, стек, таблицю розбору і вихідний потік.».

Крок 3. Виводиться завдання: «Буфер містить».

Варіанти відповіді:

- послідовність символів граматики з \$ на дні;
- вхідний рядок, за яким йде правий кінцевий маркер \$ – ознака кінця рядка;
- початковий символ граматики безпосередньо над символом \$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Буфер містить вхідний рядок, за яким йде правий кінцевий маркер \$ – ознака кінця рядка».

Крок 4. Виводиться завдання: «Стек містить».

Варіанти відповіді:

- послідовність символів граматики з \$ на дні;
- вхідний рядок, за яким йде правий кінцевий маркер \$ – ознака кінця рядка;
- початковий символ граматики безпосередньо над символом \$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Стек містить послідовність символів граматики з \$ на дні».

Крок 5. Виводиться завдання: «Спочатку стек містить».

Варіанти відповіді:

- послідовність символів граматики з \$ на дні;
- вхідний рядок, за яким йде правий кінцевий маркер \$ – ознака кінця рядка;
- початковий символ граматики безпосередньо над символом \$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Спочатку стек містить початковий символ граматики безпосередньо над символом \$».

Крок 6. Виводиться завдання: «Таблиця розбору – це двовимірний масив $M(A,a)$, де A –».

Варіанти відповіді:

- термінал;
- нетермінал;
- символ \$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Таблиця розбору – це двовимірний масив $M(A,a)$, де A – нетермінал; a – термінал або символ \$».

Крок 7. Виводиться завдання: «Таблиця розбору – це двовимірний масив $M(A,a)$, де $a \rightarrow$ ».

Варіанти відповіді:

- термінал;
- нетермінал;
- символ \$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Таблиця розбору – це двовимірний масив $M(A,a)$, де A – нетермінал; a – термінал або символ \$».

Крок 8. Виводиться пояснення:

«Аналізатор керується програмою, що працює у такий спосіб. Програма розглядає X – символ на вершині стека і a – поточний вхідний символ. Ці два символи визначають дію аналізатора. Є три можливості».

Перехід на наступний крок.

Крок 9. Виводиться завдання: «Якщо $X=a=$ \$»,».

Варіанти відповіді:

- аналізатор видаляє X зі стека і просуває покажчик вхідного потоку на наступний символ;
- аналізатор зупиняється і повідомляє про успішне закінчення розбору;
- аналізатор не зупиняється і просуває покажчик вхідного потоку на наступний символ;
- програма розглядає запис $M(X,a)$ з таблиці розбору M . Цей запис являє собою або X -продукцію граматики, або запис про помилку..

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо $X=a=$ \$, аналізатор зупиняється і повідомляє про успішне закінчення розбору».

Крок 10. Виводиться завдання: «Якщо $X=a \neq$ \$»,».

Варіанти відповіді:

- аналізатор видаляє X зі стека і просуває покажчик вхідного потоку на наступний символ;
- аналізатор зупиняється і повідомляє про успішне закінчення розбору;
- аналізатор не зупиняється і просуває покажчик вхідного потоку на наступний символ;
- програма розглядає запис $M(X,a)$ з таблиці розбору M . Цей запис являє собою або X -продукцію граматики, або запис про помилку..

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо $X=a \neq \$$, аналізатор видаляє X зі стека і просуває покажчик вхідного потоку на наступний символ.».

Крок 11. Виводиться завдання: «Якщо X – нетермінал,».

Варіанти відповіді:

- аналізатор видаляє X зі стека і просуває покажчик вхідного потоку на наступний символ;
- аналізатор зупиняється і повідомляє про успішне закінчення розбору;
- аналізатор не зупиняється і просуває покажчик вхідного потоку на наступний символ;
- програма розглядає запис $M(X,a)$ з таблиці розбору M . Цей запис являє собою або X -продукцію граматики, або запис про помилку..

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо X – нетермінал, програма розглядає запис $M(X,a)$ з таблиці розбору M . Цей запис являє собою або X -продукцію граматики, або запис про помилку. Якщо, наприклад, $M(X,a)=\{X \rightarrow UVW\}$, аналізатор замінює X на вершині стека на WVU (з U на вершині). Будемо вважати, що аналізатор на виході просто друкує використану продукцію. Якщо $M(X,a)=error$, аналізатор викликає підпрограму аналізу помилок».

Крок 12. Виводиться пояснення: «При побудові предиктивного аналізатора корисними виявляються дві функції, зв'язані з граматикою G . Ці функції, FIRST і FOLLOW, дозволяють побудувати таблицю предиктивного розбору для G , якщо, звичайно, це можливо. Множини, що породжуються цими функціями, можуть, крім того, бути використані при відновленні після помилок.».

Перехід на наступний крок.

Крок 13. Виводиться завдання: «Нехай a – довільний рядок символів граматики. Функція $\text{FIRST}(a)$ – це».

Варіанти відповіді:

- це множина терміналів, з яких починаються рядки, виведені з a ;
- множина терміналів a , що можуть з'явитися безпосередньо праворуч від A у деякій сентенціальній формі, тобто множина терміналів a таких, що існує породження вигляду $S\alpha \Rightarrow Aa\beta$ для деяких α і β ;
- це множина нетерміналів, з яких починаються рядки, виведені з a ;
- множина нетерміналів a , що можуть з'явитися безпосередньо праворуч від A у деякій сентенціальній формі.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Функція $\text{FIRST}(a)$ – це множина терміналів, з яких починаються рядки, виведені з a ».

Крок 14. Виводиться завдання: «Якщо $a \xRightarrow{*} \varepsilon$, то ε ».

Варіанти відповіді:

- також належить $\text{FOLLOW}(a)$;
- не належить $\text{FIRST}(a)$;
- також належить $\text{FIRST}(a)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо $a \xRightarrow{*} \varepsilon$, то ε також належить $\text{FIRST}(a)$ ».

Крок 15. Виводиться завдання: «Для побудови $\text{FIRST}(X)$ для всіх символів X граматики застосуємо такий алгоритм.

Крок 1:».

Варіанти відповіді:

- Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$.
- Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i=1, 2, \dots, k$, то додати ε до $\text{FIRST}(X)$.
- Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$.
- Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Крок 1. Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$ ».

Крок 16. Виводиться завдання: «Для побудови $\text{FIRST}(X)$ для всіх символів X граматики застосуємо такий алгоритм.

Крок 2:».

Варіанти відповіді:

- Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$.
- Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i=1, 2, \dots, k$, то додати ε до $\text{FIRST}(X)$.
- Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$.
- Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Крок 2. Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.».

Крок 17. Виводиться завдання: «Для побудови $\text{FIRST}(X)$ для всіх символів X граматики застосуємо такий алгоритм.

Крок 3:».

Варіанти відповіді:

- Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$.
- Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i = 1, 2, K, k$, то додати ε до $\text{FIRST}(X)$.
- Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$.
- Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Крок 3. Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$. Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i = 1, 2, K, k$, то додати ε до $\text{FIRST}(X)$.».

Крок 18. Виводиться завдання: «Для побудови $\text{FIRST}(X)$ для всіх символів X граматики застосуємо такий алгоритм.

Крок 4:».

Варіанти відповіді:

- Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$.

- Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i=1,2,K,k$, то додати ε до $\text{FIRST}(X)$.
- Якщо X – термінал, то $\text{FIRST}(X) = \{X\}$; якщо X – нетермінал, то $\text{FIRST}(X) = \{\}$.
- Якщо є продукція $X \rightarrow \varepsilon$, то додати ε до $\text{FIRST}(X)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Крок 3. Якщо X – нетермінал і є продукція $X \rightarrow Y_1 Y_2 \dots Y_k$, то включити a у $\text{FIRST}(X)$, якщо для деякого a $\text{FIRST}(Y_i)$ та ε належить усім множинам $\text{FIRST}(Y_1), \dots, \text{FIRST}(Y_{i-1})$, тобто $Y_1 K Y_{i-1} \Rightarrow \varepsilon$. Якщо ε належить $\text{FIRST}(Y_i)$ для всіх $i=1,2,K,k$, то додати ε до $\text{FIRST}(X)$ ».

Крок 19. Виводиться завдання: «Наприклад, усе, що належить $\text{FIRST}(Y_1)$ ».

Варіанти відповіді:

- належить також і $\text{FIRST}(X)$.
- не належить $\text{FIRST}(X)$.
- належить також і $\text{FIRST}(Y_2)$.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Наприклад, усе, що належить $\text{FIRST}(Y_1)$ належить також і $\text{FIRST}(X)$ ».

Крок 20. Виводиться завдання: «Якщо з Y_1 не виводиться ε , то».

Варіанти відповіді:

- нічого більше не додаємо до $\text{FIRST}(X)$.
- додаємо $\text{FIRST}(Y_1)$.
- додаємо $\text{FIRST}(Y_2)$ і т.д.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо з Y_1 не виводиться ε , то ми нічого більше не додаємо до $\text{FIRST}(X)$ ».

Крок 21. Виводиться завдання: «Якщо $Y_1 \Rightarrow \varepsilon$, то».

Варіанти відповіді:

- нічого більше не додаємо до $\text{FIRST}(X)$.
- додаємо $\text{FIRST}(Y_1)$.
- додаємо $\text{FIRST}(Y_2)$ і т.д.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Якщо $Y_1 \Rightarrow \varepsilon$, то додаємо $\text{FIRST}(Y_2)$ і т.д.».

Крок 22. Виводиться завдання: «Тепер FIRST для будь-якого рядка $X_1X_2K X_n$ можна обчислити у такий спосіб.».

Послідовність:

1. Вважаємо $\text{FIRST}(X_1X_2K X_n) = \{\}$.
2. Додамо до $\text{FIRST}(X_1X_2K X_n)$ усі не ε -символи $\text{FIRST}(X_1)$.
3. Нарешті, додамо ε до $\text{FIRST}(X_1X_2K X_n)$, якщо для всіх i $\text{FIRST}(X_i)$ містить ε .
4. Додамо також всі не ε -символи з $\text{FIRST}(X_2)$, якщо $\varepsilon \in \text{FIRST}(X_1)$, всі не ε -символи з $\text{FIRST}(X_3)$, якщо ε належить як $\text{FIRST}(X_1)$, так і $\text{FIRST}(X_2)$, і т.д.

Якщо вибрано правильну відповідь, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку: «Вважаємо $\text{FIRST}(X_1X_2K X_n) = \{\}$. Додамо до $\text{FIRST}(X_1X_2K X_n)$ усі не ε -символи $\text{FIRST}(X_1)$. Додамо також всі не ε -символи з $\text{FIRST}(X_2)$, якщо $\varepsilon \in \text{FIRST}(X_1)$, всі не ε -символи з $\text{FIRST}(X_3)$, якщо ε належить як $\text{FIRST}(X_1)$, так і $\text{FIRST}(X_2)$, і т.д. Нарешті, додамо ε до $\text{FIRST}(X_1X_2K X_n)$, якщо для всіх i $\text{FIRST}(X_i)$ містить ε .».

Крок 23. Виводиться завдання: «Приклад. Розглянемо граматику арифметичних виразів:

$$E \rightarrow TE', E' \rightarrow +TE' | \varepsilon,$$

$$T \rightarrow FT', T' \rightarrow * FT' | \varepsilon,$$

$$F \rightarrow (E) | id$$

Запишемо функції FIRST .».

Для неї заповніть комірки:

- FIRST(E) = FIRST(T) = FIRST(F) = {_____};
- FIRST(E') = {_____};
- FIRST(T') = {_____};

Якщо вибрано правильну відповідь, то виводиться повідомлення про завершення, пропонується повторне тестування. Інакше – відображається повідомлення про помилку:

$$\llbracket \text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{(\text{id})\};$$

$$\text{FIRST}(E') = \{+, \varepsilon\};$$

$$\text{FIRST}(T') = \{*, \varepsilon\}; \gg.$$

3.3 Блок-схема алгоритму тренажера

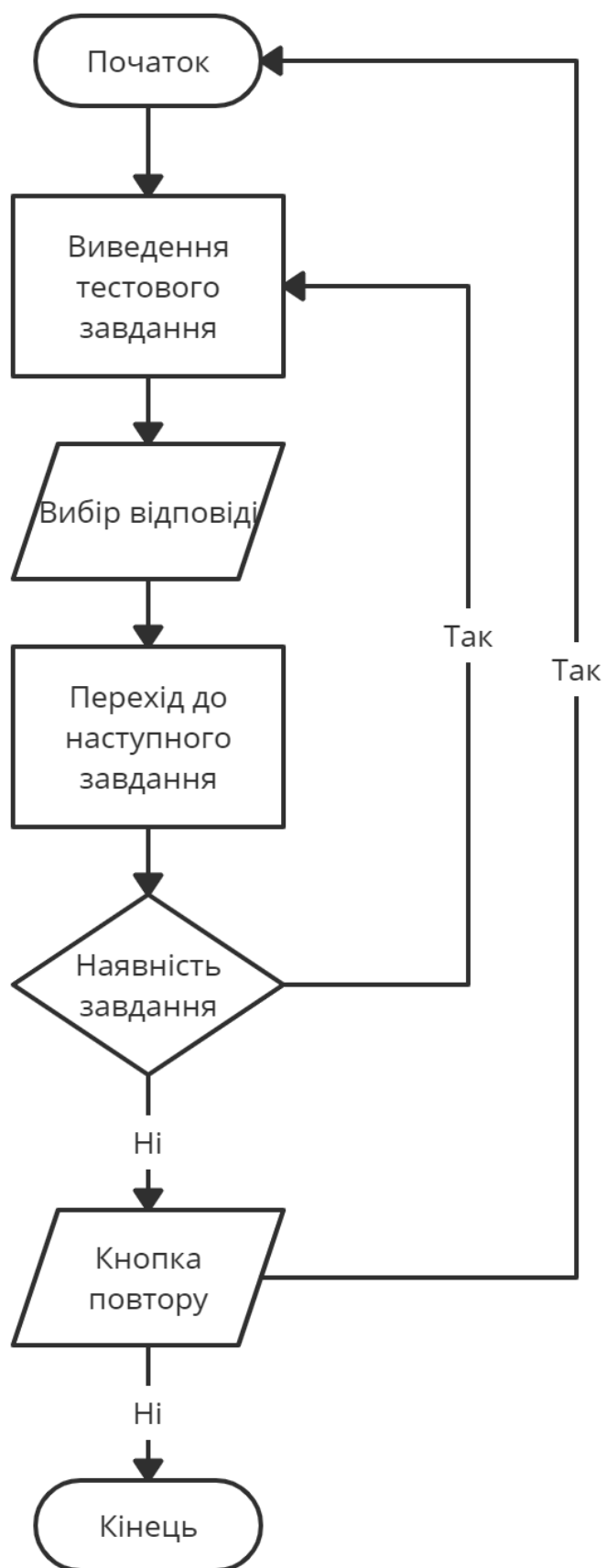


Рисунок 3.1 – Блок-схема роботи тренажера

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмної реалізації

Елементи у вікні проекту розташовано наступним чином:

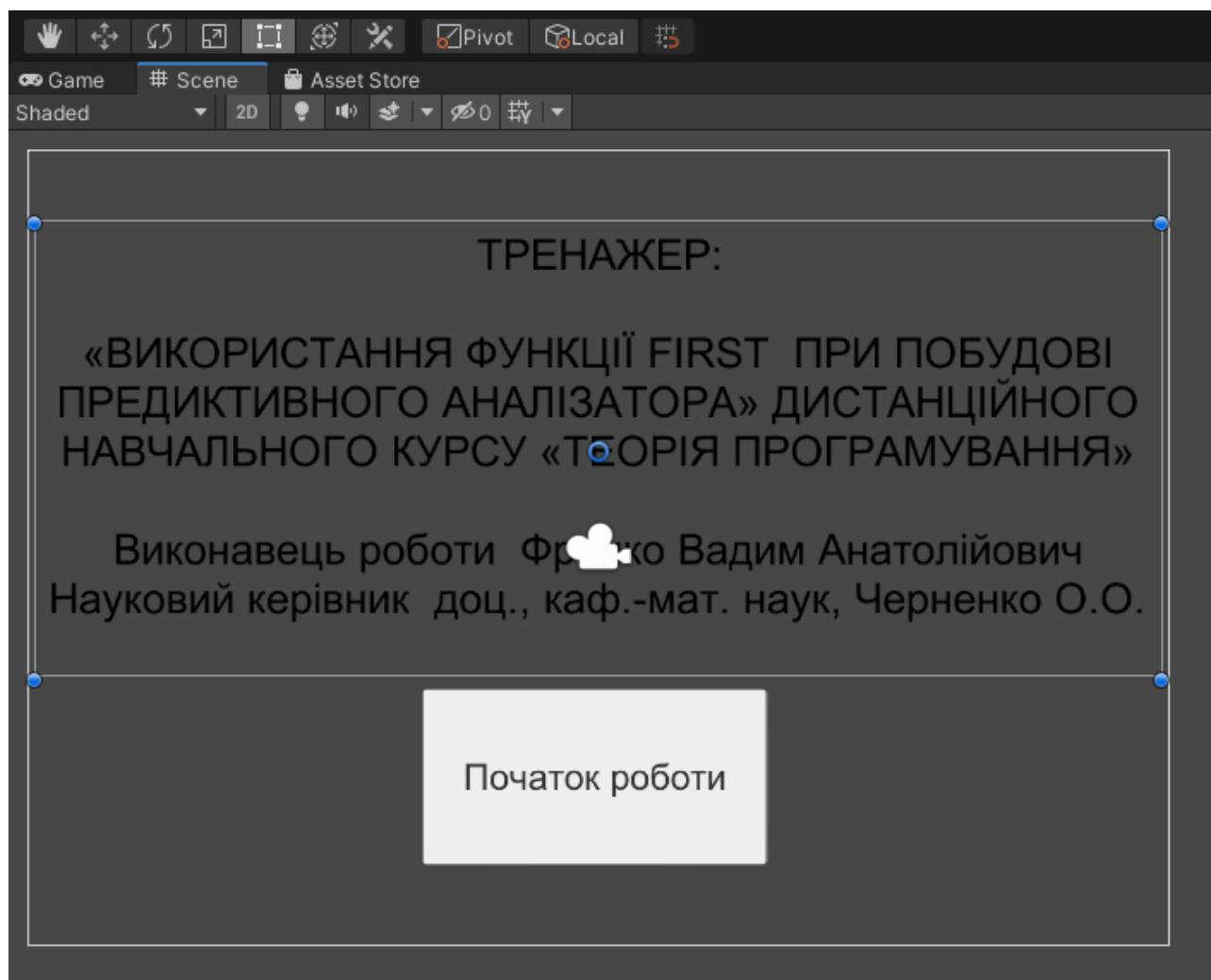


Рисунок 4.1 – Розташування елементів у вікні проекту

Всі інтерактивні елементи в тренажері додано за допомогою вбудованих функцій платформи Unity. Всі скрипти додано за допомогою середовища для написання програмного коду.

Для роботи з текстом було використано текстовий скрипт, робота в даному скрипті реалізована за прикладом роботи з офісними програмами.

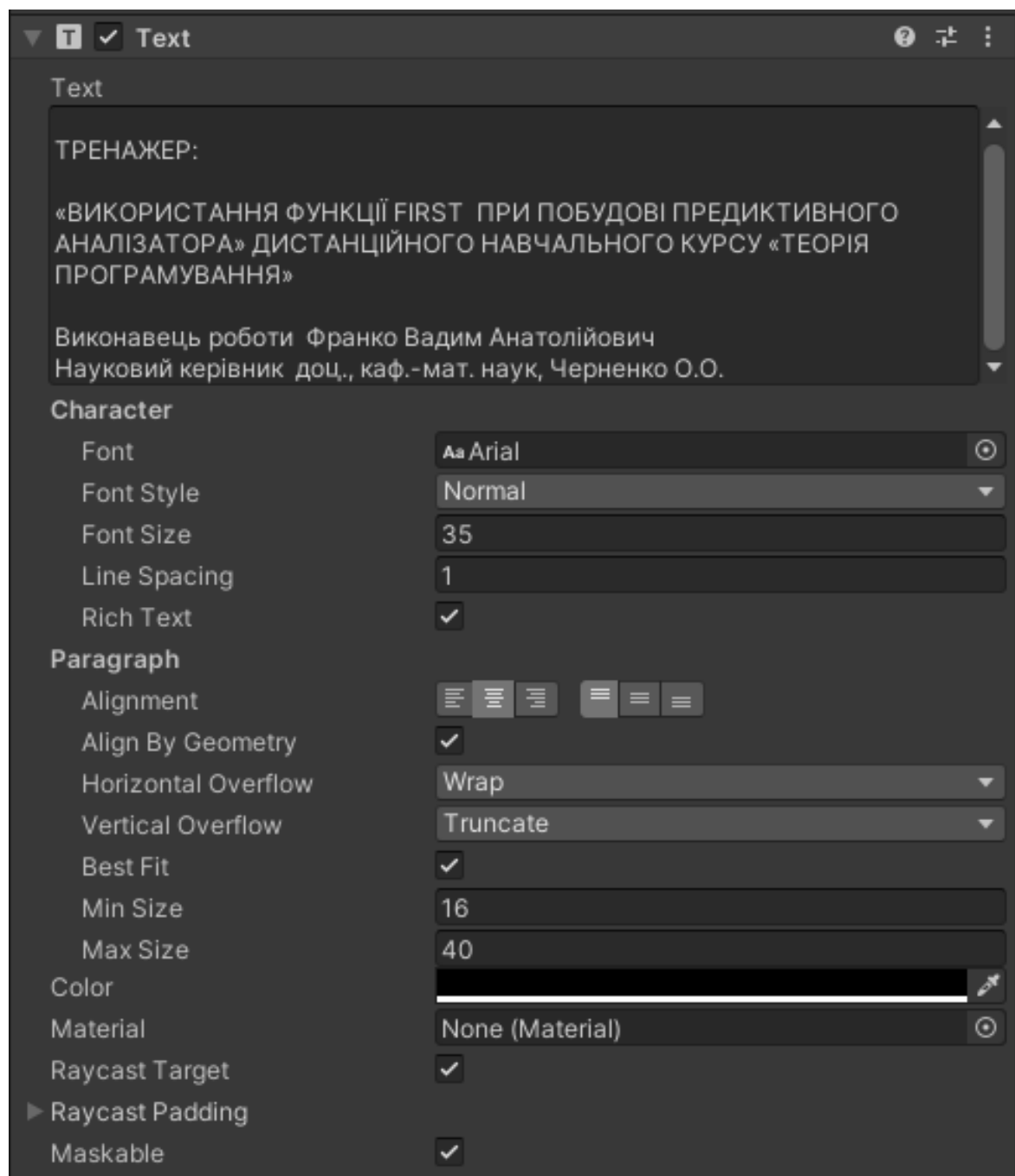


Рисунок 4.2 – Текстовий скрипт

Для роботи з кнопкою для початку роботи було використано наступний скрипт для роботи з кнопками.

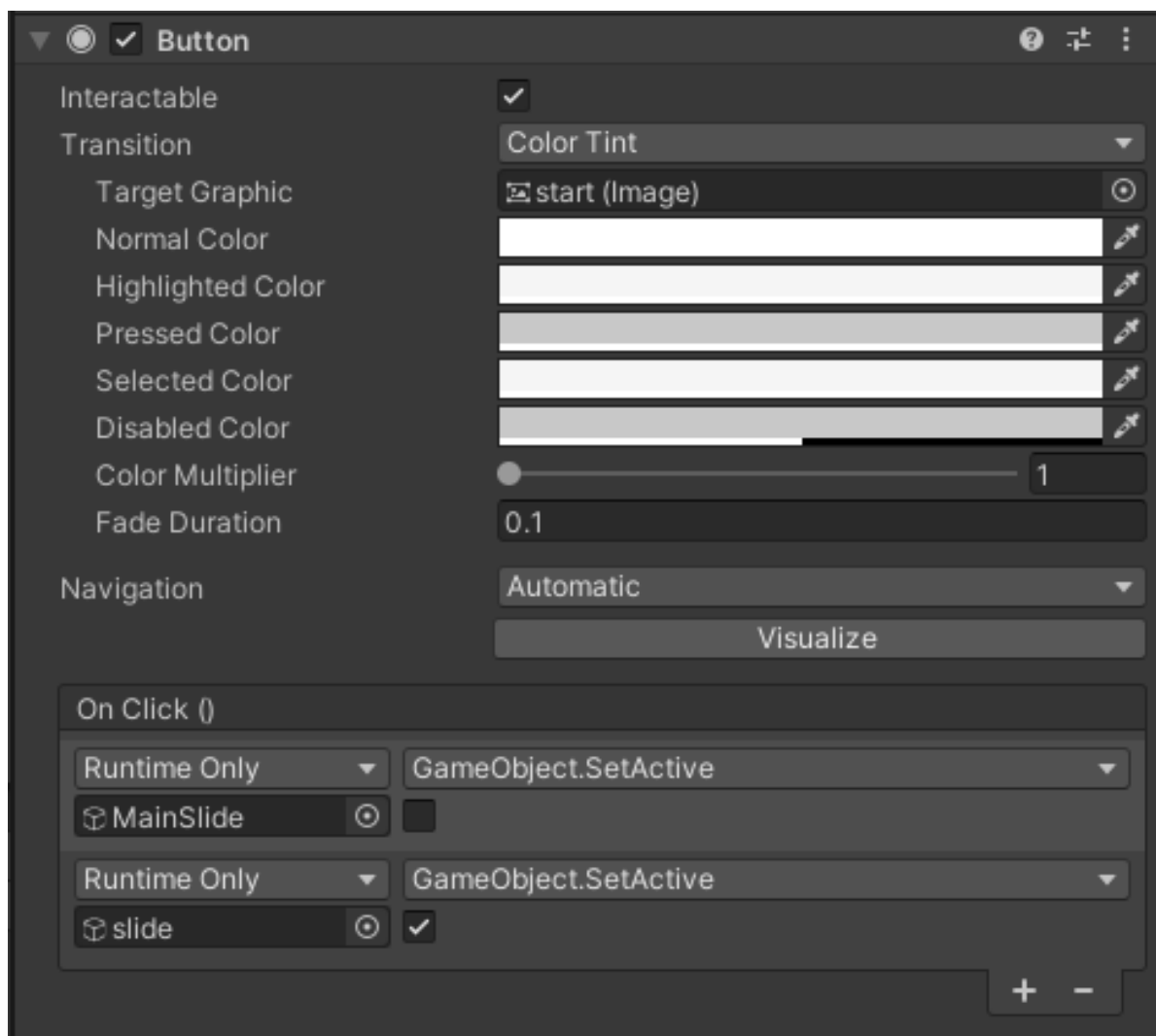


Рисунок 4.3 – Скрипт кнопки для початку роботи

Під час роботи з тестовими завданнями для роботи з кнопкою з правильним варіантом відповіді було використано наступний скрипт для роботи з кнопками.

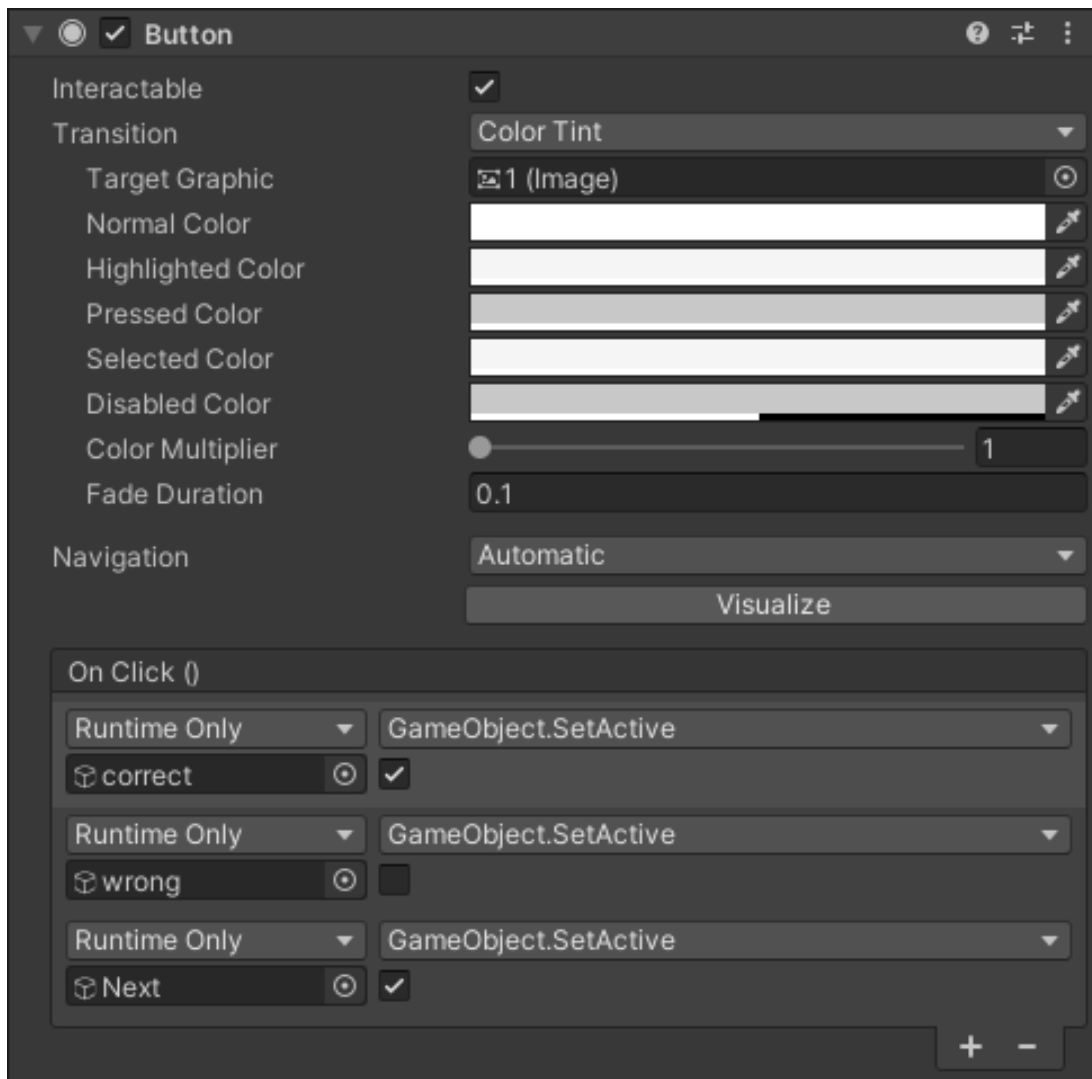


Рисунок 4.4 – Скрипт кнопки з правильним варіантом відповіді

Під час роботи з тестовими завданнями для роботи з кнопкою з неправильним варіантом відповіді було використано наступний скрипт для роботи з кнопками.

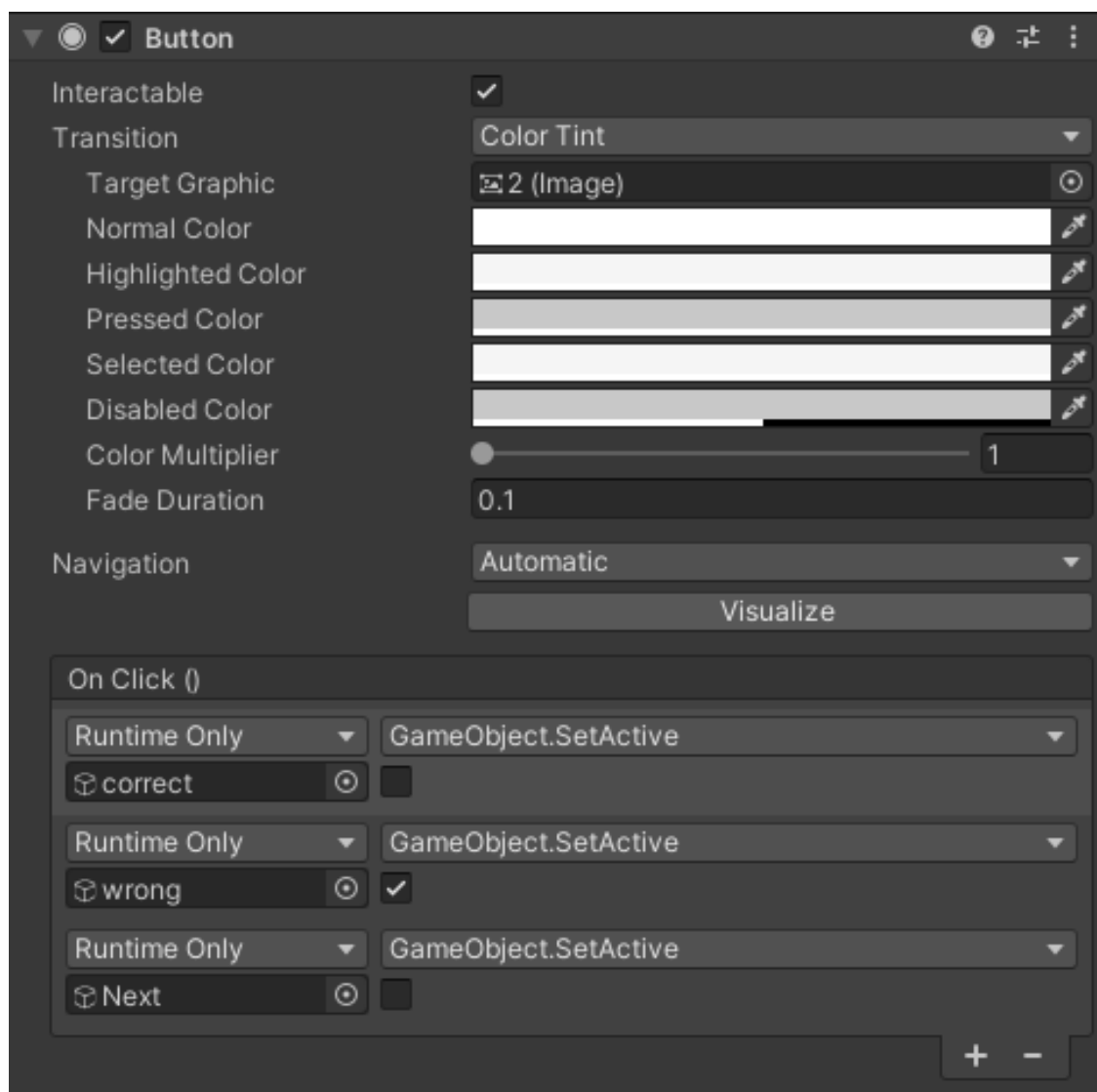


Рисунок 4.5 - Скрипт кнопки з неправильним варіантом відповіді

4.2 Інструкція для роботи з тренажером

Після запуску тренажеру користувач переходить до наступного вікна:

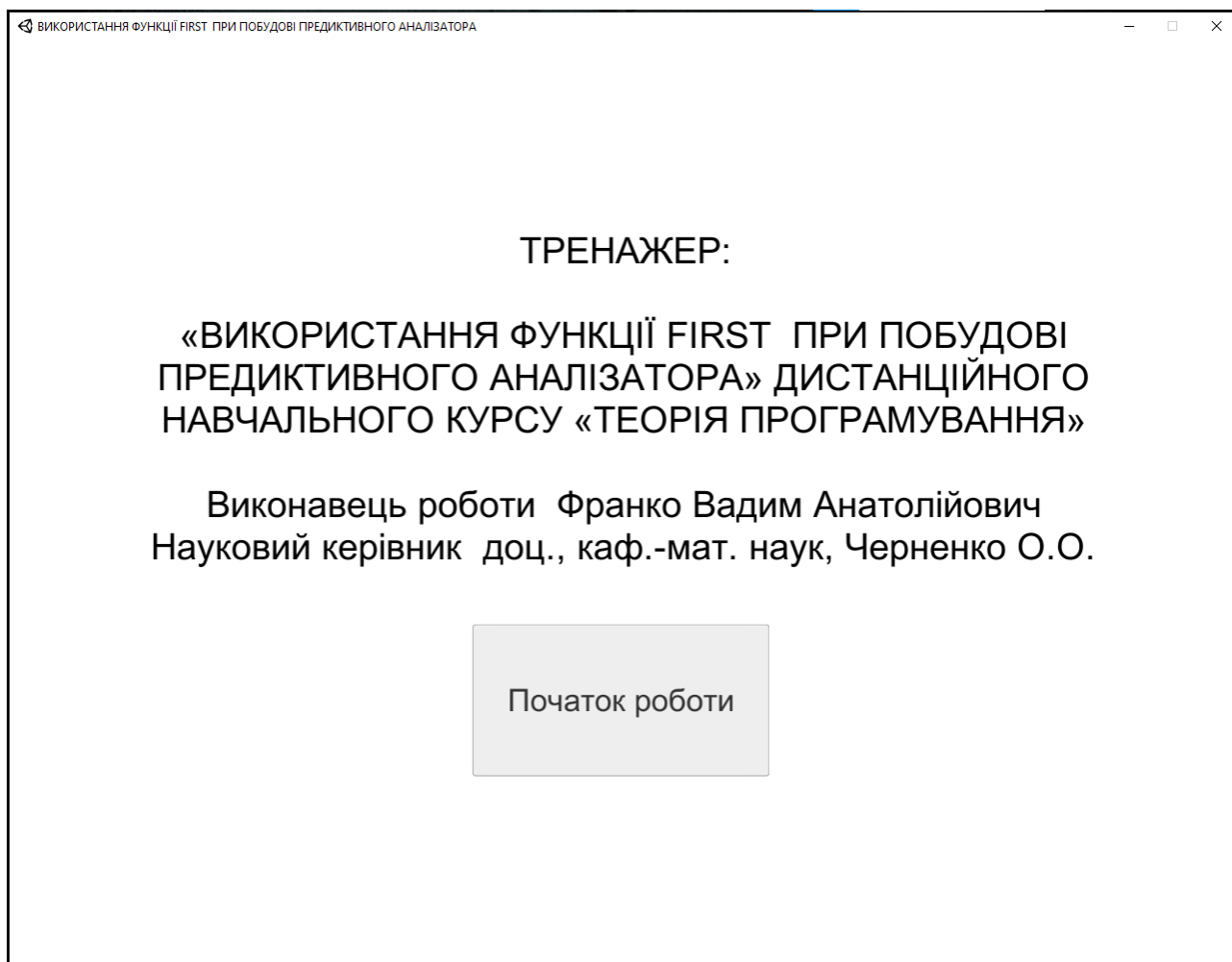


Рисунок 4.6 – Початок роботи з тренажером

Видача теоретичної інформації реалізована наступним чином:

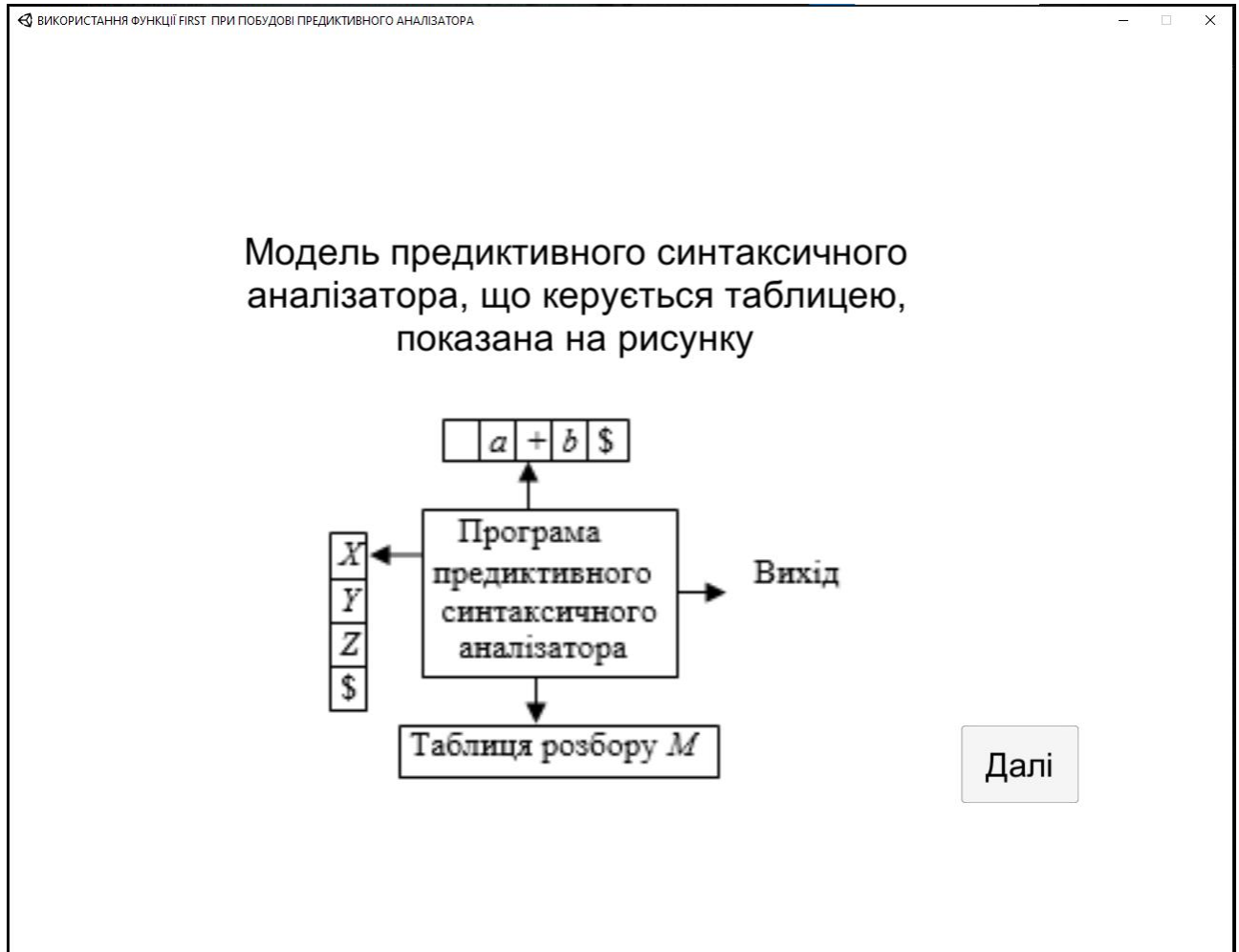


Рисунок 4.7 – Видача теоретичної інформації

Практичні завдання реалізовано наступним чином:

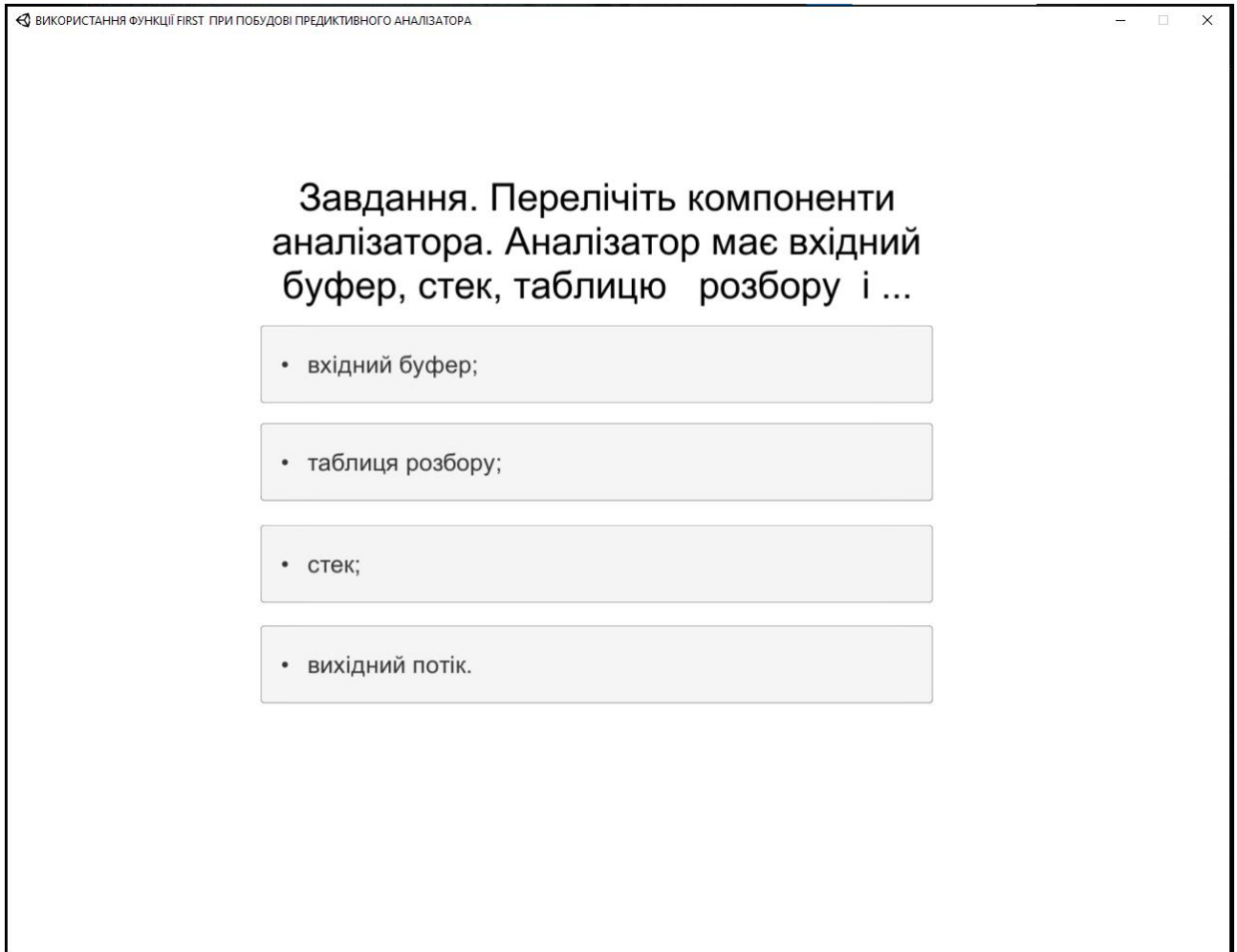


Рисунок 4.8 – Видача практичного завдання

Після вибору неправильної відповіді видача відповідного повідомлення з підказкою.

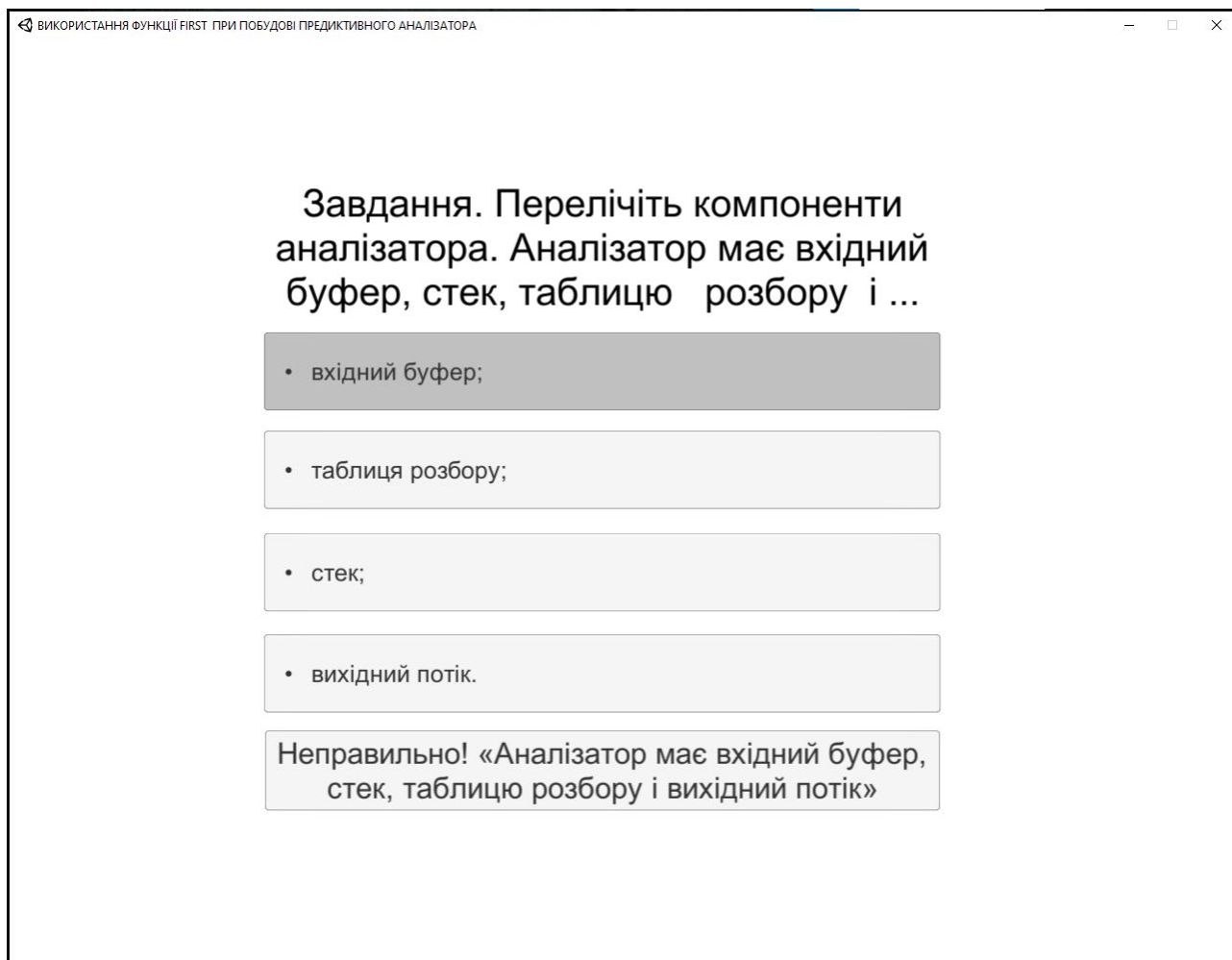


Рисунок 4.9 – Практичне завдання з неправильною відповіддю

Після вибору правильної відповіді видача відповідного повідомлення та продовження роботи.

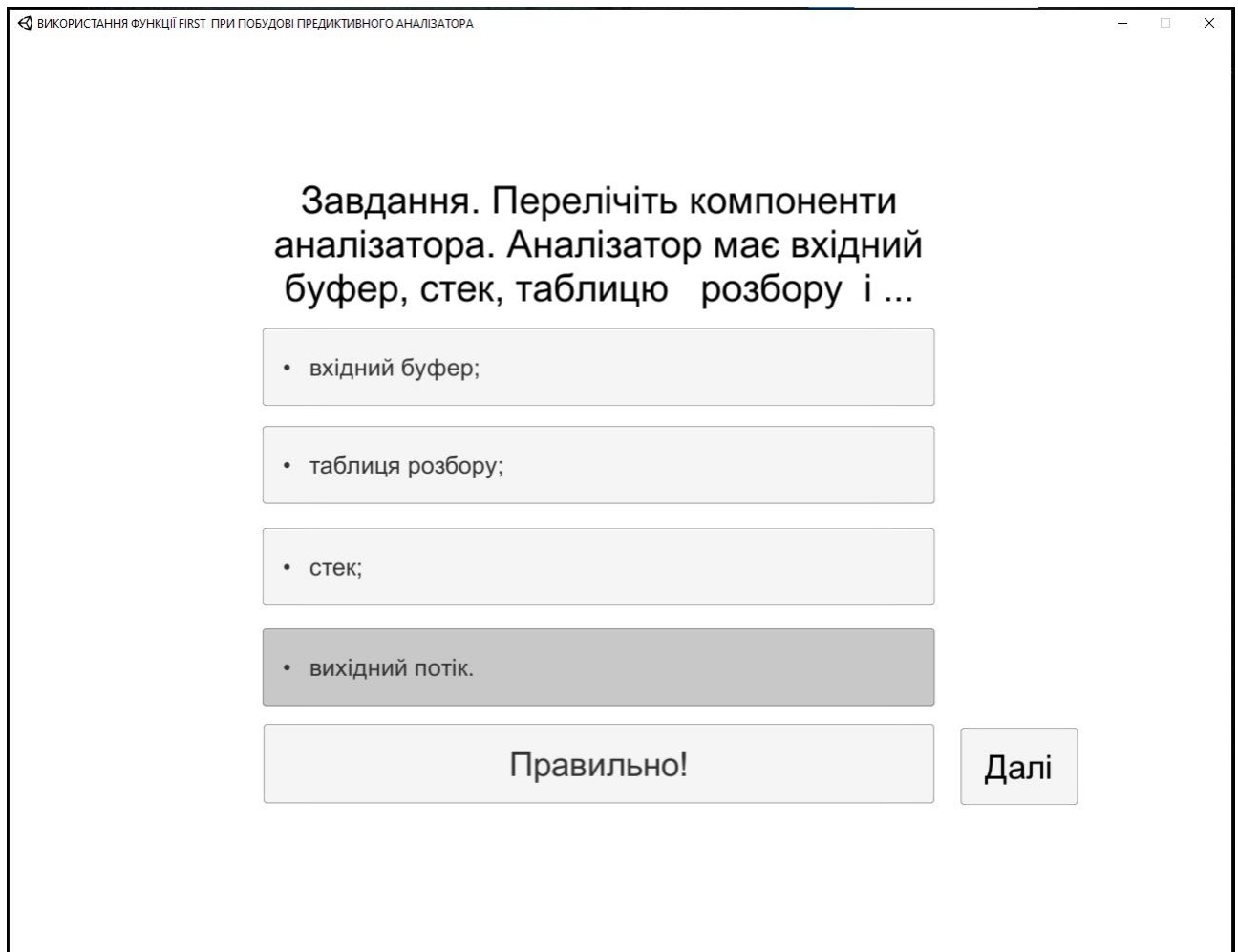


Рисунок 4.10 – Практичне завдання після вибору правильного варіанту відповіді

Видача теоретичного матеріалу до теми:

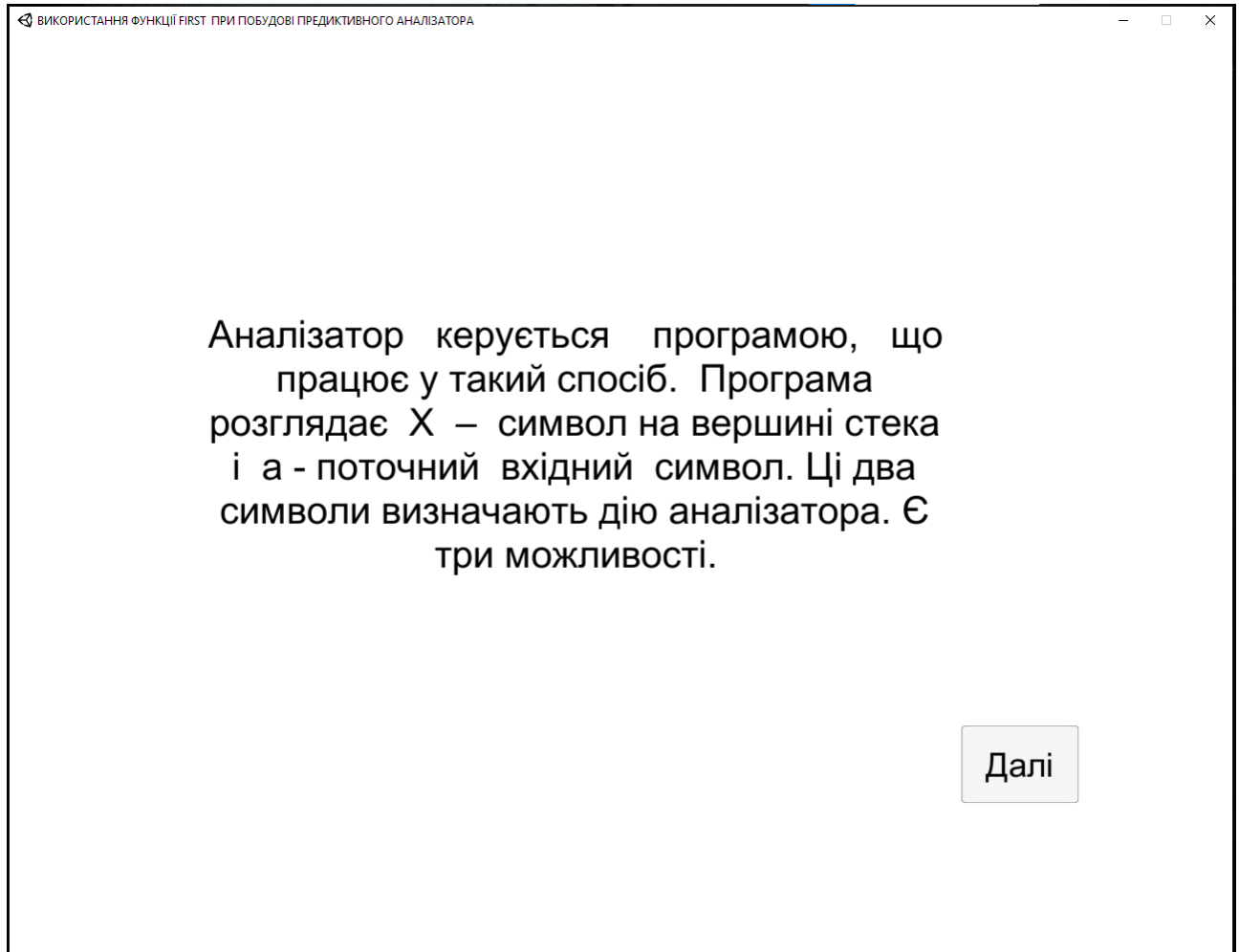


Рисунок 4.11 – Теоретичний матеріал до теми

Видача підказки реалізована наступним чином. Для закриття підказки необхідно натиснути на повідомлення.

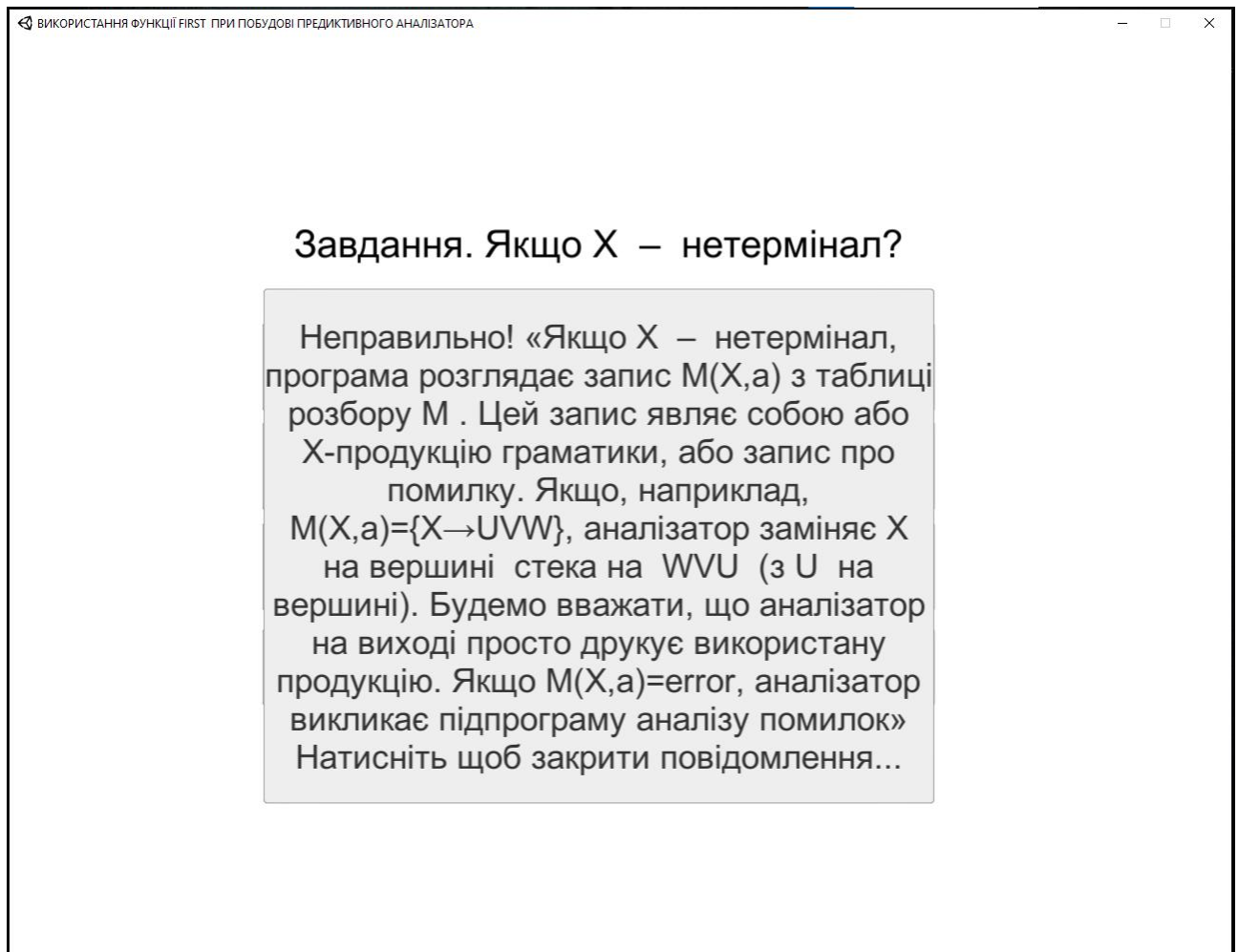


Рисунок 4.14 – Повідомлення про помилку в завданні

Можливо змінити відповідь після отримання підказки.

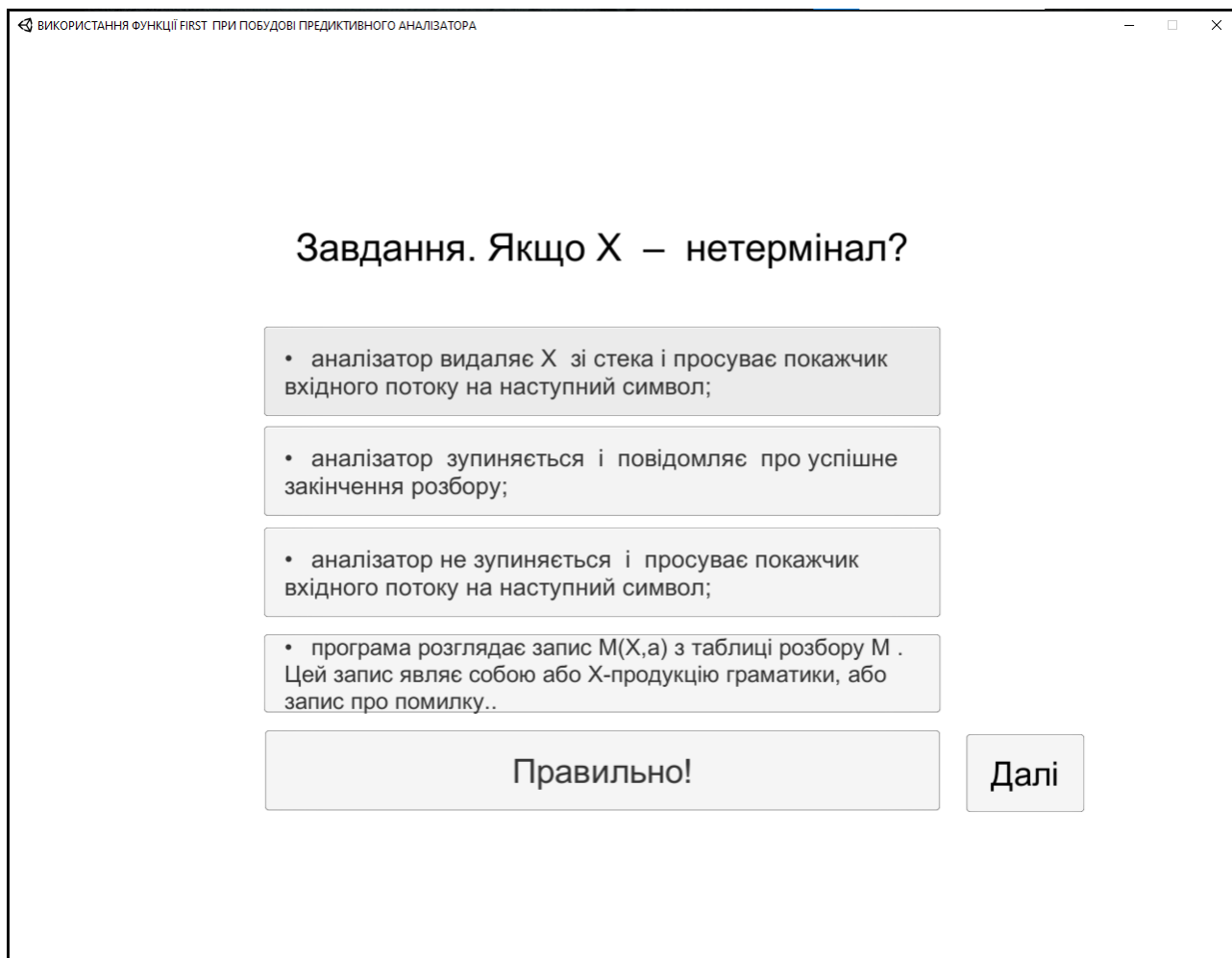


Рисунок 4.15 – Зміна відповіді, продовження роботи

Практичне завдання з заповненням пустих клітинок виглядає наступним чином:

Використання функції FIRST при побудові предиктивного аналізатора

Приклад. Розглянемо граматiku арифметичних виразів:

$$\begin{aligned} E &\rightarrow TE', E' \rightarrow +TE' | \epsilon, \\ T &\rightarrow FT', T' \rightarrow *FT' | \epsilon, \\ F &\rightarrow (E) | id \end{aligned}$$

Запишемо функції FIRST - ...

- $\text{FIRST}(E) = \text{FIRST}(T) = \text{FIRST}(F) = \{\text{ } \};$
 - $\text{FIRST}(E') = \{\text{ } \};$
 - $\text{FIRST}(T') = \{\text{ } \};$

1- "id" 2- "ε" 3- "*, ε"

1- "(,id" 2- "+, ε" 3- "*, ε"

Правильно!

Далі

Рисунок 4.16 – Завдання з заповненням пустих клітинок

Після завершення роботи з практичними завданнями виводиться повідомлення про завершення, пропонується повторне тестування.

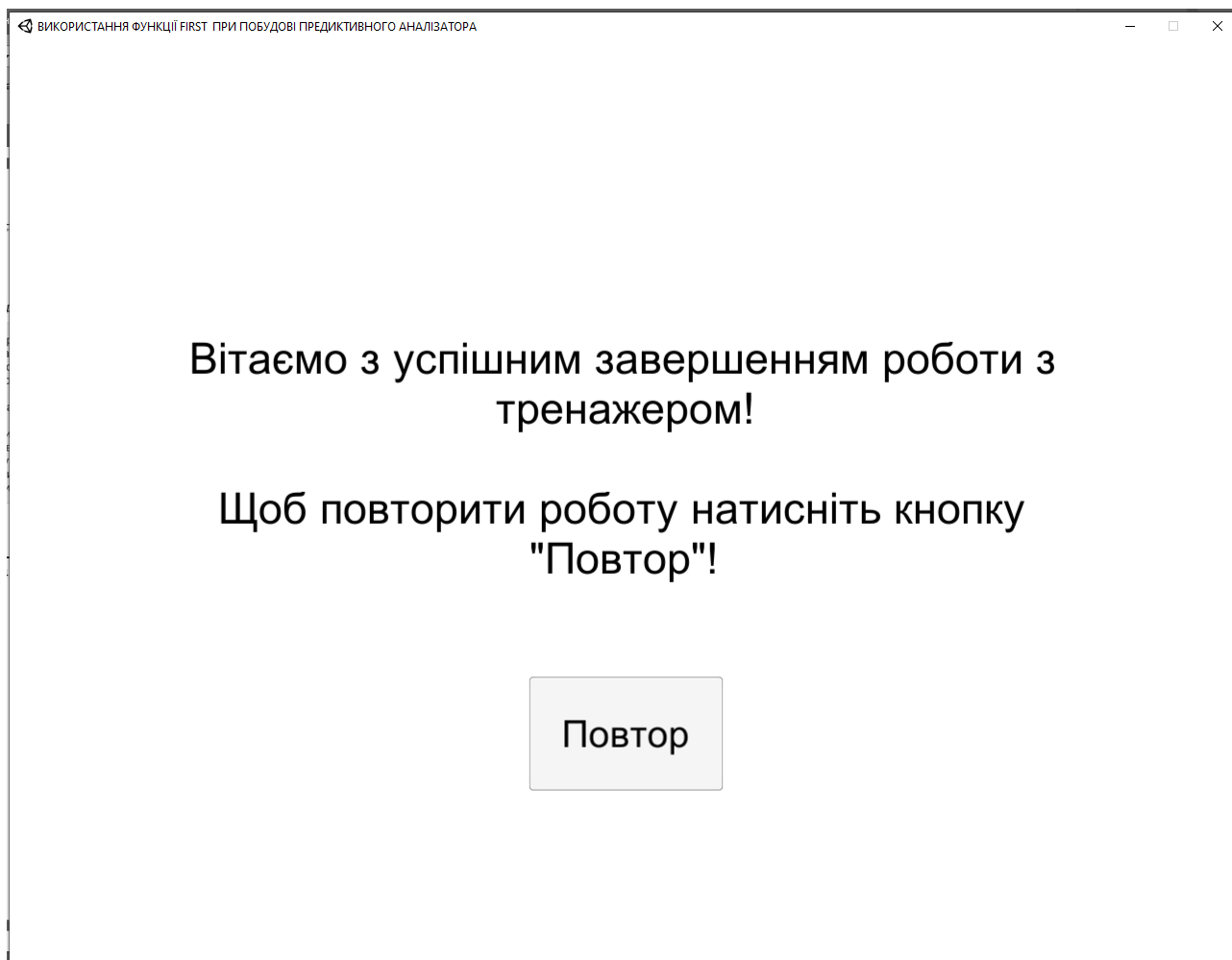


Рисунок 4.17 – Кінець роботи з тренажером

ВИСНОВКИ

Дистанційне навчання представляє собою нову організацію освіти, що ґрунтується на використанні як кращих традиційних методів отримання знань, так і нових інформаційних та телекомунікаційних технологій, а також на принципах самоосвіти. Воно призначене для широких верств населення незалежно від матеріального забезпечення, місця проживання та стану здоров'я. Дистанційне навчання дає змогу впроваджувати інтерактивні технології викладання матеріалу, здобувати повноцінну вищу освіту або підвищувати кваліфікацію і має такі переваги, як гнучкість, актуальність, зручність, модульність, економічна ефективність, інтерактивність, відсутність географічних кордонів для здобуття освіти.

Гнучкість дистанційного навчання полягає у можливості викладання матеріалу курсу з урахуванням підготовки та здібностей студентів. Це досягається створенням альтернативних сайтів для одержання більш детальної або додаткової інформації із складних тем, або низки питань-підказок. Актуальність дистанційного навчання проявляється у можливості упровадження новітніх педагогічних, психологічних і методологічних розробок з розбиттям матеріалу на окремі функціонально завершені модулі (теми), які вивчаються у міру засвоєння і відповідають здібностям окремого студента або групи загалом.

Розглянемо основні завдання роботи:

- розглянуто теоретичні відомості про використання функції FIRST при побудові предиктивного аналізатора;
- розглянуто перспективи дистанційного навчання;
- розроблено алгоритм тренажера;

Тренажер повинен передбачити можливість перевірки правильності кожного з кроків алгоритму та повідомляти про помилку у разі неправильної відповіді.

При проходженні тренажеру спочатку користувачеві повинні відображатися теоретичні завдання, а вже потім приклади.

При розробці програмного продукту пропонується використати наступну структуру:

- стартова сторінка;
- теоретичні завдання;
- приклади;
- результат тестування.

СПИСОК ЛІТЕРАТУРИ

1. Шовкопляс О.А. Звіт про науково-дослідну роботу «Моделювання навчального процесу вивчення економіко-математичних дисциплін з використанням комп'ютерних технологій» / О. А. Шовкопляс, О. О. Базиль, О. А. Літвіненко, О. С. Мазманішвілі – Суми: Вид-во СумДУ, 2015. – 53 с.
2. Карпенко М.М. Розвиток дистанційного навчання як відповідь на сучасні виклики для України /М.М. Карпенко // Стратегічні пріоритети.–№4(33).– 2014
3. Мясковська М.О. Комп'ютерне моделювання як ефективний метод посилення міждисциплінарних зв'язків / М.О. Мясковська // Збірник наукових праць Кам'янець-Подільського національного університету імені Івана Огієнка. Серія педагогічна / [редкол.: П.С. Атаманчук (голов. ред.) та ін.]. – Кам'янець- Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, 2014. – Вип. 20: Управління якістю підготовки майбутнього вчителя фізико-технологічного профілю. – С. 289-291.
4. Дистанційний курс «Теорія програмування» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКИ «Полтавський університет економіки і торгівлі»». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=2009>
5. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
6. NP Chunking (State of the art). Association for Computational Linguistics. – Режим доступу: [https://aclweb.org/aclwiki/index.php?title=NP_Chunking_\(State_of_the_art\)](https://aclweb.org/aclwiki/index.php?title=NP_Chunking_(State_of_the_art))
7. Середовище розробки. – Режим доступу: <https://visualstudio.microsoft.com/ru/downloads/>
8. Платформа для розробки. – Режим доступу: <https://store.unity.com/ru/>

9. Робота з платформою для розробки. – Режим доступу: unity.com/ru/

ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ